

Interactive comment on “A numbering algorithm for finite elements on extruded meshes which avoids the unstructured mesh penalty” by Gheorghe-Teodor Bercea et al.

Gheorghe-Teodor Bercea et al.

gb308@doc.ic.ac.uk

Received and published: 15 September 2016

We would like to thank the second reviewer for the suggestions to improving the manuscript. We have addressed all the questions and comments individually.

Comment 1: *I assume that λ defined as number of extruded layers stands for the number of vertices of a vertical mesh, rather than the number of segments in the vertical. In this case, looking at Equation 5: d_2 can be either 0 (for vertices) or 1 (for segments). Then the statement $0 \leq l \leq \lambda - d_2$ will let λ go out of bounds, unless the second \leq symbol is replaced by $<$. Similar observation for l -loop in Algorithm 3. In case this assumption is wrong, it should be made more clear what is meant with layers.*

Printer-friendly version

Discussion paper



Answer: λ is in fact the number of segments in the vertical. We have explicitly defined λ accordingly and ensured that the usage is consistent throughout the manuscript.

Comment 2: *The title suggests that besides a memory layout for function spaces, also a numbering strategy in the horizontal would be discussed.*

Answer: We do not think that the title implies this, however, we have changed the title in response to reviewer 1. We hope that this title is less ambiguous in what it is suggesting the contributions of the paper are. We feel that the contributions of the paper are expressly stated and the title does not read contrary to those claims.

Comment 3: *Looking at Figure 3., how is the numbering of $n + \#$ related to $m + \#$, and possibly nodes internal to the triangle? Are $m + \#$ and $n + \#$ related to different function spaces?*

Answer: The nodes internal to the triangle would be numbered in a similar way, but we feel that this would unnecessarily complicate the figure. Because the horizontal mesh is unstructured, there is no simple relationship between m and n , however our results show that using a suitable ordering of the horizontal mesh (such that m and n are typically "close") is important for performance. A given function space will have degrees of freedom associated with one or more entity types, for example a continuous cubic space in the horizontal combined with a continuous linear mesh in the vertical would have the degrees of freedom shown in the figure, plus one degree of freedom per horizontal facet.

Comment 4: *Given the title I would have expected to see discussed what would be the impact (on e.g. performance) to number $n + \#$ (as in Figure 3) first in vertical fashion (zig-zag up-down, rather than right-left).*

Answer: Reviewer 1 also had a similar comment. We acknowledge that the ordering in each entity column is not unique. We do not see an obvious advantage to zig-zag up-down as opposed to left-right numbering. We have added some text to section 3.2 in

[Printer-friendly version](#)[Discussion paper](#)

this vein. We note that we do not believe the title claims that this is the best numbering algorithm on extruded meshes, merely an approach which works well.

Comment 5: *It would help to see a visualisation of a practical numbering for a mesh of a few triangles and few levels, for a few function function space configurations, besides Algorithm 1.*

Answer: We do not believe this would add clarity to the paper. For example, the newly added Figure 3 merely numbers the topological entities on a single extruded triangle and is already complex.

Comment 6: *In Algorithm 1, first “dofsfs” is assigned with round brackets, later with square brackets.*

Answer: Thanks, fixed.

Comment 7: *In Algorithm 1, the second last line makes reference to l , which is invalid outside the vertical loop.*

Answer: Thanks, fixed.

Comment 8: *In Algorithm 1, perhaps exchanging loops l and d_2 can avoid the last 2 lines by looping l in $0, 1, \dots, \lambda-1-d_2$? (possibly without changing the resulting order)*

Answer: Unfortunately this would change the resulting numbering.

Comment 9: *In Algorithm 3, can I assume that for a single DG-DG cell-entity, $(dof_0, dof_1, \dots, dof_{k-1})$ are contiguous?*

Answer: Yes.

Comment 10: *In Algorithm 3, subscript fs in $Lfs(v)$ missing*

Answer: Thanks, fixed.

Comment 11: *In Algorithm 3, d_2 unassigned, should be referenced as subscript of V ?*

Answer: Fixed by making d_2 an explicit input to the algorithm.

Comment 12: *In Figure 5, almost none of the results have reached as mentioned the “performance plateau” with 100 layers for the badly ordered mesh. It would be interesting to see how many layers are required for all discretizations to reach this plateau.*

Answer: The reviewer is correct that in the badly ordered case almost none of the results have reached the performance plateau, we have reworded the discussion to address this. We do not believe that extending the experiment until we achieve some plateau in the “bad” case would be of practical interest since typical simulations have tens, not hundreds of layers in the vertical.

Comment 13: *Is there an expected benefit in going higher order to reach the plateau with less layers?*

Answer: There is indeed an expected benefit in going to high order, we have added a note to this effect at the end of the discussion.

Interactive comment on Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-140, 2016.

Printer-friendly version

Discussion paper

